Package Tracker

Team Members:
Chase Grajeda(grajec)
Jianye Peng (pengj6)
Eddie Poon (poone)
Aneesh (koluka)

Feedback

- More specified testing strategy
- Modified connection in between objects
- Optimized AI component

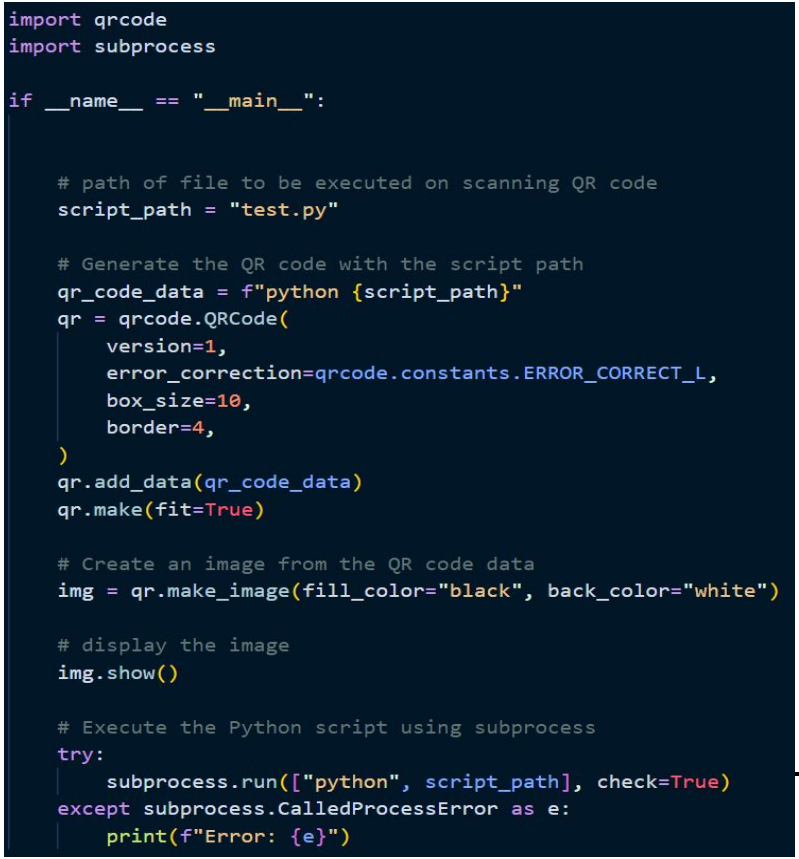
Testing:

- Unit Testing:
  Ensure that oracle, AI, smart contract is working on their end.
- Simulation:
  Make dummy test cases to simulate the delivery.
- Error handling:
  Improper input from senders(e.g. non-existing address)
  Recipient forgets to scan and close the contract
- Performance:
  Request to send 2000 package

AI

- Debugging PyTorch data loaders
- Created lookup table for street IDs
- Model features:
    - Traffic density is a value between 0 and 1000. To improve learning, this feature will be normalized with log normalization
    - Some data points have a sudden drop off in traffic density (to 0), showing an error in data collection. These points will be given a small amount of noise to prevent bias
- Hoping to start training next week and over the break

Blockchain

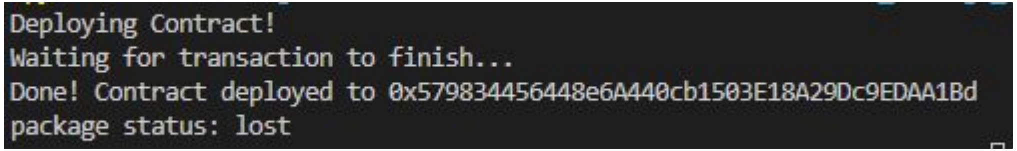


```python
import qrcode
import subprocess

if __name__ == "__main__":


    # path of file to be executed on scanning QR code
    script_path = "test.py"

    # Generate the QR code with the script path
    qr_code_data = f"python {script_path}"
    qr = qrcode.QRCode(
        version=1,
        error_correction=qrcode.constants.ERROR_CORRECT_L,
        box_size=10,
        border=4,
    )
    qr.add_data(qr_code_data)
    qr.make(fit=True)

    # Create an image from the QR code data
    img = qr.make_image(fill_color="black", back_color="white")

    # display the image
    img.show()

    # Execute the Python script using subprocess
    try:
        subprocess.run(["python", script_path], check=True)
    except subprocess.CalledProcessError as e:
        print(f"Error: {e}")
```

- Created python program to generate QR code.
- The QR code can be linked with another python file that utilizes web3 to call the solidity functions in the smart contract.
- As of now the QR code and image has been successfully generated.
- Will write python code to update delivery status similar to the demo mentioned in the presentation.
- Blockchain code now emits events to store crucial information which will be helpful for testing.
- We still need to work on linking the AI and blockchain. We are planning on doing this once we are able to make sure the QR code scanning works with the smart contract without any problem.

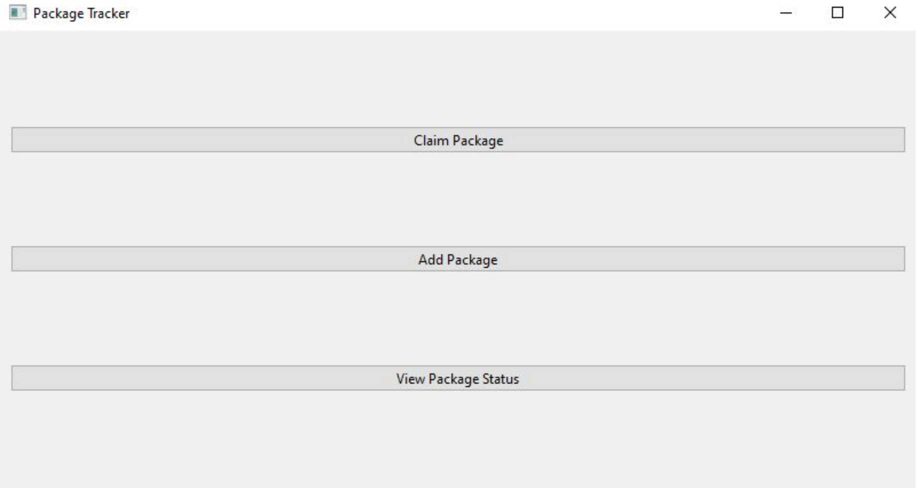
Integration:

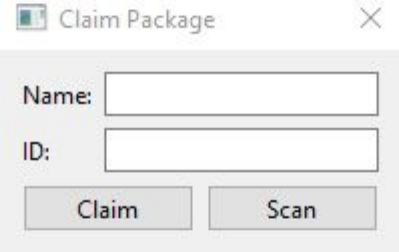
- Using web3.py, compile Solidity code in python, using ganache for personal blockchain testing
- Able to call functions from the solidity contract in python directly, even supply arguments
- AI is built in python
- Use PyQt as a GUI wrapper

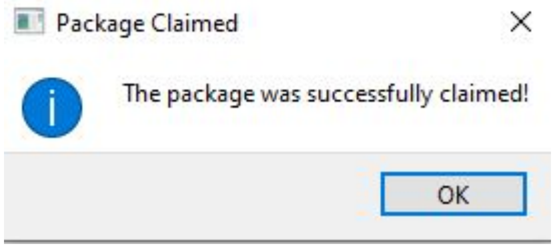


```
Deploying Contract!
Waiting for transaction to finish...
Done! Contract deployed to 0x579834456448e6A440cb1503E18A29Dc9EDAA1Bd
package status: lost
```

Demo:







The package was successfully claimed!

Q & A